UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/722,294	11/25/2003	Barrie Richard Timpe	SYS 049 PA	9093

29673 7590 05/10/2007
STEVENS & SHOWALTER LLP
7019 CORPORATE WAY
DAYTON, OH 45459-4238

EXAMINER	
NAM, HYUN	
ART UNIT	**PAPER NUMBER**
2109	
MAIL DATE	**DELIVERY MODE**
05/10/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	**Application No.** 10/722,294	**Applicant(s)** TIMPE ET AL.
	Examiner Hyun Nam	**Art Unit** 2109

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)□ Responsive to communication(s) filed on _____.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-29* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-29* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☒ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date *2/27/04 & 6/20/05*.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .
5) □ Notice of Informal Patent Application
6) □ Other: _____.

The following is a quotation of the second paragraph of 35 U.S.C. 112:

> The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

Claim 7 is rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

Claim 7 recites the limitation "said data element" in line 3. There is insufficient antecedent basis for this limitation in the claim.

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1-25 and 27-29 are rejected under 35 U.S.C. 102(b) as being anticipated by Fung (U.S. Patent Number 4,535,420).

With respect to claim 1, Fung teaches a queuing system comprising:

a queue (see Figure 1 Controller Input Buffer RAM) having a plurality of addressable storage locations (RAM) associated therewith;

queue logic (see Figure 1) to control write operations to said queue (see Figure 4 Computer Bus 15), said queue logic operatively configured to write data events (see Figure 1 MUX 17, Device Address PROM 18, Device Address Configuration Switch, IUAX signal, DRY/FRY signal, Bus 15, and Computer 16) to said queue in a re-circulating sequential manner (see Column 5 Line 67 and Column 6 Line 1-2) irrespective of whether previously stored data has been read out (see Column 6 Line 2-5);

a current event counter (see Figure 1 RAM Access Counter 13) updated by said queue logic to keep track of a count value that corresponds to the total number of data events written (see Column 8 Line 63-65) to said queue, said current event counter capable of counting an amount (see Column 2 Line 22-24; Note, a 16x16 bit memory configuration requires 16 bit counter) that is greater than said plurality of addressable storage locations (see Column 4 Line 38; Note, a 16 bit counter is capable of exceeding number of addressable storage in four 4x16 memory configurations);

read logic (see Figure 1 RAM Access Counter 13, RAM Address Holding Register 12, Tri-State Drivers 14, MUX 11, Bus 21, Bus 15, and Computer 16) operatively configured to read data events from said queue according to a prescribed manner (see Abstract Line 8-11), said read logic further communicably coupled to said current event counter for reading said count value stored therein (see Column 3 Line 41-42); and

a read pointer (see Figure 1 MUX 11 Output) controlled by said read logic that relates to where in said queue data is to be read from said queue, wherein said read logic can read from said queue based upon said read pointer independently of write operations (see Figure 1 MUX 17, Device Address PROM 18, Device Address Configuration Switch, IUAX signal, DRY/FRY signal, Bus 15, and Computer 16) to said queue (see Column 3 63-65).

With respect to claim 2, Fung teaches the queuing system according to claim 1, further comprising a previous event counter (see Figure 1 RAM Access Counter 13) that is controlled by said read logic that relates to a previous value of said current event counter (see Figure 1 RAM Access Counter 13; Note, the Counter 13 counts previous and current events) at the time of a previous read operation on said queue by said read logic.

With respect to claim 3, Fung teaches the queuing system according to claim 2, wherein said read logic is further operatively configured to determine whether data has been lost in the queue due to queue overflow (see Column 3 Line 50-58) based upon a comparison of a current value (see Column 3 Line 45-46) of said current event counter and said previous value (see Column 3 Line 40-43) stored in said previous event counter.

With respect to claim 4, Fung teaches the queuing system according to claim 2, wherein said read logic is further operatively configured to perform read operations according to a first prescribed manner (see Column 3 Line 52-53) when no queue overflow is detected (see Column 3 Line 47-51), and said read logic performs read operations from said queue according to a second prescribed manner (see Column 3 Line 56-58) when queue overflow is detected (see Column 3 Line 53-55).

With respect to claim 5, Fung teaches the queuing system according to claim 2, wherein said read pointer is derived directly from a predetermined number of the lowest order bits of said previous event counter (see Figure 2 Counter 13 Q_A-Q_D output).

With respect to claim 6, Fung teaches the queuing system according to claim 1, wherein a write pointer (see Figure 1 MUX 11 Output) is derived directly from a predetermined number of the lowest order bits of said current event counter (see Figure 2 Counter 13 Q_A-Q_c output; Note, lowest 3 bits derives count value).

With respect to claim 7, Fung teaches the queuing system according to claim 1, wherein said queue logic stores for each data event, a combination of a sequence number derived from the value of said current event counter value (see Figure 2 Counter 13, CK Input) and said data element (see Figure 2 Counter 13 A, B, C, D Input tied to GND or logic zero) in said queue.

With respect to claim 8, Fung teaches the queuing system according to claim 1, wherein said user communicates with said queue through an intermediate interface (see Figure 1 Computer 16).

With respect to claim 9, Fung teaches the queuing system according to claim 1, wherein said read logic is associated with a plurality of users, each user comprising:

read logic operatively configured to read information from said queue according to a prescribed manner, said read logic further communicably coupled to said current event counter for reading said count value stored therein; and

a read pointer updated by said read logic that relates to where in said queue data is to be read from said queue, wherein said read logic can read from said queue based upon said read pointer independently of write operations to said queue, wherein said read pointer and read logic of each user (see Column 2 Line 40-45; Note, Microprocessor and Minicomputer) operates independently of one another (see Figure 1 Buses 15, 20, and 21).

With respect to claim 10, Fung teaches the queuing system according to claim 1, wherein said read logic further cascades data events read from said queue into a local queue for subsequent processing (see Figure 4 Scratch Pad Memory 404).

With respect to claim 11, Fung teaches a queuing system comprising:

a queue (see Figure 1 Controller Input Buffer RAM) having a predetermined number of addressable storage locations (RAM);

an event counter (see Figure 1 RAM Access Counter 13) operatively configured to sequentially update (see Figure 2 Counter 13 CK input) a count value (see Figure 2 Counter 13 Q_a, Q_b, Q_c, and Q_d output) stored therein each time a data event (Note, read or write event) is written into said queue, said count value capable of storing a maximum count (see Column 2 Line 22-24; Note, 16x16 bit memory configuration requires 16 bit counter) that exceeds said predetermined number of addresses (see Column 4 Line 38; Note, a 16 bit counter is capable of exceeding number of addressable storage in 4x16 memory configuration);

a write pointer (see Figure 1 MUX 17 output) that is derived from said count value stored in said event counter from which a select addressable storage location (see Controller Input Buffer RAM 10) of said queue can be determined for temporarily storing each data event that is to be queued (see Figure 1 Device Address PROM 18);

a read pointer (see Figure 1 MUX 11 output) from which a desired addressable storage location (see Figure 1 Controller Input Buffer RAM) of said queue can be identified for a read operation (see Figure 1 RAM Access Counter 13, RAM Address Holding Register 12, Tri-State Drivers 14, MUX 11, Bus 21, Bus 15, and Computer 16);

queue logic (see Figure 1) communicably coupled to said queue, said event counter, and said write pointer to control writing new data events (see Figure 4 RAM 10) to said queue; and

read logic (see Figure 1 RAM Access Counter 13, RAM Address Holding Register 12; Tri-State Drivers 14, MUX 11, Bus 21, Bus 15, and Computer 16) coupled to said queue, said event counter and said read pointer, said read logic operatively configured to read from said queue in a first manner (see Column 3 Line 52-53) when no overflow of said queue is detected (see Column 3 Line 47-51), and to read from said queue in a second manner (see Column 3 Line 56-58) when overflow of said queue is detected (see Column 3 Line 53-55).

With respect to claim 12, Fung teaches the queuing system, according to claim 11, wherein said write pointer is encoded into said event counter (see Figure 2 Counter 13; Note, all of the inputs to the counter and coupled logic devices to the inputs).

With respect to claim 13, Fung teaches the queuing system according to claim 11, wherein said predetermined number of addresses of said queue is defined by the expression: $m = 2^n$, where m is the number of addresses (see Figure 4 RAM 10, 16 slots), and n is a positive integer (see Figure 2 four RAMs).

With respect to claim 14, Fung teaches the queuing system according to claim 13, wherein said write pointer is defined by the lowest n bits of said event counter (see Figure 2 Counter 13 Q_A-Q_C output; Note, lowest 3 bits defines count value).

With respect to claim 15, Fung teaches the queuing system according to claim 14, wherein a portion of said count value is stored in said queue with each data event written thereto, said portion defined by at least one bit of said count value starting at the n+1 bit (see Figure 2 Counter 13 Q_D output; Note, the 4th bit Q_D defines count value).

With respect to claim 16, Fung teaches the queuing system according to claim 11, wherein each read from the queue is nondestructive, and each write to the queue overwrites the current content of the storage location addressed by the write pointer (see Column 6 Line 48-60; Note, each write operation reused the one slot available in the RAM and there was no need for deleting the content of the slot when it will be rewritten).

With respect to claim 17, Fung teaches the queuing system according to claim 11, wherein said first manner of reading from said queue comprises reading from said queue in a first in, first out manner (see Column 6 Line 48-60; Note, FIFO with queue depth of one) such that a list sequential read follows a temporal aging from the oldest to the newest events in said queue and said second manner of reading from said queue comprises reading from said queue in a manner that reads the most recent events written to the queue first (see Column 6 Line 48-60; Note, read occurs soon as it is written).

With respect to claim 18, Fung teaches the queuing system according to claim 11, wherein said second manner of reading from said queue comprises setting said read pointer to said write pointer prior to initiating a read operation (see Column 6 Line 48-60; Note, it is inherent that RAM 10 is addressed before read or write operation begins).

With respect to claim 19, Fung teaches the queuing system according to claim 11, wherein said second manner of reading from said queue comprises reading in a first direction (see Column 3 Line 55-58) for a predetermined portion of a read cycle, and reading in a second direction (see Column 3 Line 58-62) for a remainder portion of said read cycle.

With respect to claim 20, Fung teaches the queuing system according to claim 11, wherein said second manner of reading from said queue comprises setting said read pointer to a position a predetermined number of addresses in a direction opposite to said write pointer, and beginning a read cycle wherein a plurality of data events are read in the direction of write operations to said queue (see Column 10 Line 50-55).

With respect to claim 21, Fung teaches the queuing system according to claim 11, wherein said read logic further comprises a previous event counter that keeps track of a representation of said count value of said event counter at the time of a previous read operation (see Column 3 Line 44-47).

With respect to claim 22, Fung teaches the queuing system according to claim 21, wherein said read counter is derived from the lowest order bits of said previous event counter (see Figure 2 Counter 13 Q_A-Q_C output; Note, lowest 3 bits derives count value).

With respect to claim 23, Fung teaches a method of queuing data comprising:

defining a queue (see Figure 1 Controller Input Buffer RAM) having addressable storage locations (RAM) associated therewith;

keeping track of a current count value (see Figure 1 RAM Access Counter, Tri-State Drivers 14, and Bus 20; Note, current count value available at the Bus 20) that corresponds to the total number of data events written to said queue, said current count value capable of counting an amount (see Column 2 Line 22-24; Note, 16x16 bit memory configuration requires 16 bit counter) that is greater than the number of said addressable storage locations (see Column 4 Line 38; Note, a 16 bit counter is capable of exceeding number of addressable storage in 4x16 memory configuration);

keeping track of a write pointer (see Figure 1 MUX 17 output) that corresponds to a position (Address) in said queue for a write operation (see Figure 2 Gate Logic 35 and RAM 10-1 WE or Write Enable input) thereto;

writing data events (see Figure 1 MUX 17, Device Address PROM 18, Device Address Configuration Switch, IUAX signal, DRY/FRY signal, Bus 15, and Computer 16) to said queue in a re-circulating sequential manner (see Column 5 Line 67 and Column 6 Line 1-2) irrespective of whether previously stored data has been read out (see Column 6 Line 2-5); and

for each user (see Column 2 Line 40-45; Note, Microprocessor and Minicomputer) associated with said queue:

keeping track of a previous count value that corresponds to said current count value (see Column 3 Line 44-47) at the time of a previous access to said queue thereby; and

reading from said queue according to a prescribed manner (see Column 3 Line 52-53).

With respect to claim 24, Fung teaches the method according to claim 23, wherein at least one user reads from said queue according to a first prescribed manner (see 3 Column 3 Line 52-53) when no queue overflow has been detected (see Column 3 Line 47-51), and reads from said queue according to a second prescribed manner (see Column 3 Line 56-58) when overflow has been detected (see 3 Column 3 Line 53-55).

With respect to claim 25, Fung teaches the method according to claim 24, wherein queue overflow is detected if the difference between said, current count value and said previous count value is greater than a total number of said addressable storage locations (see Figure 2 Counter 13 CO or Carry Output; Note, it is inherent that counter has carry output when overflow of counter occurs which indicates that current counter exceeded its addressable storage).

With respect to claim 27, Fung teaches the method according to claim 24, wherein a select user reads from said queue comprising:

reading said current count value (see Column 3 Line 45-46);

reading said previous count value (see Column 3 Line 46-47);

comparing said current count value to said previous count value to determine whether overflow has occurred to the queue with respect to said user (see Column 3 Line 46-53);

if no overflow is detected:

reading a queued data (see Column 3 Line 62) event based upon said read pointer;

updating said read pointer based upon said first predetermined manner (see Column 3 Line 52-53);

and

updating said previous event counter value; and

if overflow is detected:

updating said read pointer based upon a second predetermined manner;

reading at least one queued data event based upon said read pointer;

updating said read pointer based upon said second predetermined manner; and

updating said previous event counter value.

With respect to claim 28, Fung teaches the method according to claim 27, further comprising after each read of said queue where no overflow is detected:

determining a new current count value (see Column 3 Line 45-46);

comparing said new current count value to said previously stored count value to determine whether overflow has occurred (see Column 3 Line 46-53); and

if overflow is detected:

updating said read pointer based upon a second predetermined manner;

reading at least one queued data event based upon said read pointer;

updating said read pointer based upon said predetermined manner; and

updating said previous event counter value.

With respect to claim 29, Fung teaches the method according to claim 27, wherein said read pointer is updated to a new position that corresponds to a current value of said write pointer (see Column 6 Line 50-55; Note, after a write event a read event occurs at same address point).

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

Claim 26 is rejected under 35 U.S.C. 103(a) as obvious over Fung in view of Smith et al. (U.S. Patent 4,872,110).

With respect to claim 26, Fung teaches the method according to claim 24, wherein a select user requests data events from said queue comprising:

specifying a total number of requested data events (see Column 6 Line 46-48);

accessing said queue to obtain said data events (see Column 6 Line 52-53); and

queuing any extracted data events in a local queue accessible by said user (see Column 6 Line 60-61).

But Fung does not teach a method wherein a select user requests data events from said queue comprising specifying a timeout period that represents the maximum amount of time said user is willing to wait for said data events.

However, Smith et al. teaches a method wherein a select user requests data event comprising specifying a timeout period (see Figure 3A Timer 68 and 74) that represents the maximum amount of time said user is willing to wait for said data events (see Column 4 Line 23).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to incorporate command acknowledge and timeout circuit of Smith et al. to the queue system of Fung. One of ordinary skill in the art would be motivated to do because keeping track of acknowledge and timeout signal will improve software throughput and operator will be immediately alerted if problem exists.

Claims 1 and 2 are rejected under 35 U.S.C. 103(a) as obvious over Unger (U.S. Patent 3,935,563) in view of Heap et al. (U.S. Patent 4,231,106).

With respect to claim 1, Unger teaches a queuing system comprising:

a queue (see Figure 2 Footprint RAM 42) having a plurality of addressable storage locations (see Column 3 Line 68) associated therewith; queue logic (see Figure 2) to control write operations (see Figure 2 Write /Count line 43) to said queue, said queue logic operatively configured to write data events to said queue in a re-circulating sequential manner (see Column 3 Line 5-7) irrespective of whether previously stored data has been read out;

a current event counter (see Figure 2 Readout Counter 62) updated by said queue logic to keep track of a count value (see Figure 2 Readout Counter 62 output) that corresponds to the total number of data events (see Figure 2 Count/Read 337 line 64) written to said queue;

read logic (see Figure 2) operatively configured to read data events (see Figure 2 Reference Updater 50) from said queue according to a prescribed manner (see Column 3 Line 12), said read logic further communicably coupled to said current event counter for reading said count value stored therein (see Figure 2 Circle Counter 44 output); and

a read pointer (see Figure 3 Selector 54 output) controlled by said read logic that relates to where in said queue data is to be read from said queue (see Figure 2 Footprint RAM 42 and Z Register 30), wherein said read logic can read from said queue based upon said read pointer independently of write operations (Note, the write operation does not influence the read operation) to said queue.

But Unger does not teach, a queuing system comprising a current event counter capable of counting an amount that is greater than said plurality of addressable storage locations.

However, Heap et al. teaches a queuing system comprising a current event counter capable of counting an amount that is greater than said plurality of addressable storage locations (see Abstract 3-5; Note, it is inherent that the prescribed instruction sequence executed in prescribed functions are capable of exceeding any given queue size in infinite test loop scenario).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to incorporate counter technique of Heap et al. to a queuing or memory footprint system of Unger. One of ordinary skill in the art would be motivated to do because the counting apparatus can produce a record of the occurrence of various software events or actions in the data processor (see Column 2 Line 60-62).

With respect to claim 2, the modification teaches the queuing system according to claim 1, further comprising a previous event counter (see Unger's Figure 2 Circle Counter 44) that is controlled by said read logic that relates to a previous value of said current event counter (see Heap et al.'s Figure 1 Timing and Counting Apparatus 12) at the time of a previous read operation on said queue by said read logic.

Claim 1 is rejected under 35 U.S.C. 103(a) as obvious over Unger (U.S. Patent 3,935,563) in view of Lindsay's publication, *A Hardware Monitor Study of a CDC KRONOS System.*

With respect to claim 1, Unger teaches a queuing system comprising:

a queue (see Figure 2 Footprint RAM 42) having a plurality of addressable storage locations (see Column 3 Line 68) associated therewith; queue logic (see Figure 2) to control write operations (see Figure 2 Write /Count line 43) to said queue, said queue logic operatively configured to write data events to said queue in a re-circulating sequential manner (see Column 3 Line 5-7) irrespective of whether previously stored data has been read out;

a current event counter (see Figure 2 Readout Counter 62) updated by said queue logic to keep track of a count value (see Figure 2 Readout Counter 62 output) that corresponds to the total number of data events (see Figure 2 Count/Read 337 line 64) written to said queue;

read logic (see Figure 2) operatively configured to read data events (see Figure 2 Reference Updater 50) from said queue according to a prescribed manner (see Column 3 Line 12), said read logic further communicably coupled to said current event counter for reading said count value stored therein (see Figure 2 Circle Counter 44 output); and

a read pointer (see Figure 3 Selector 54 output) controlled by said read logic that relates to where in said queue data is to be read from said queue (see Figure 2 Footprint RAM 42 and Z Register 30), wherein said read logic can read from said queue based upon said read pointer independently of write operations (Note, the write operation does not influence the read operation) to said queue.

But Unger does not teach, a queuing system comprising a current event counter capable of counting an amount that is greater than said plurality of addressable storage locations.

However, Lindsay teaches a current event counter capable of counting an amount that is greater than said plurality of addressable storage locations (see Page 136 Section II.A.1 Line 16).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of Applicant's invention to incorporate counter technique of Lindsay to a queuing or memory footprint system of Unger. One of ordinary skill in the art would be motivated to do because the counting apparatus can produce a record of the occurrence of various software events or actions in the data processor by multiple of users. This with the memory footprint readout will accomplish comprehensive analysis of the given software (see Page 136 Section II.A.1 Line 20-24).

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Hyun Nam whose telephone number is (512) 270-1725. The examiner can normally be reached on Monday through Friday from 8:30 AM to 5:00 PM. If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, David Robertson, can be reached on 571-272-4186. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

HN
Hyun Nam
(571) 270-1725
7 May 2007

DAVID L. ROBERTSON
SUPERVISORY PATENT EXAMINER